

2. Le diagramme de classes UML

La documentation

M2104 - Object Oriented Design & Modelling

2021 Edition

Rappels : le diagramme de classes

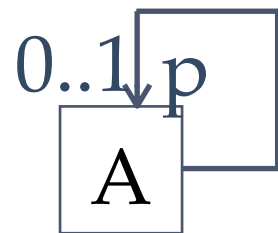
- ❖ Les **objets** dénotent des **entités**, possèdent des **opérations** et des **données** (des méthodes et des attributs). Ils sont définis par les **classes**.
- ❖ Les objets peuvent faire référence à d'autres objets dans des **associations** et appeler des méthodes d'autres classes dans des **liens d'usage**. Le but du diagramme de classes UML est de visualiser ces relations entre classes et le contenu des classes.
- ❖ Les multiplicités plurielles dénotent l'utilisation de **conteneurs** en code Java: tableaux, listes, ensembles, tableaux associatifs. Une **classe-association** dans le diagramme de classes représente les caractéristiques d'une association entre un couple d'objets par un troisième objet, ce qui peut être implémenté par un tableau associatif.

Compléments sur le diagramme de classes

- * **La navigabilité** est la direction de l'information dans une association, une navigabilité à sens unique signifiant qu'un objet fait référence à un autre sans que la réciproque soit vraie.



- * Les **associations réflexives** peuvent être utilisées quand un objet fait référence à un autre objet de la même classe.



D'autres informations sont généralement omises mais peuvent être ajoutées si nécessaires : visibilités sur les rôles des associations, notes *ordered* ou *set* sur les multiplicités, ou notes en texte libre.

Documentation du code

- ❖ Nécessaire : l'utilisation de code ne doit pas impliquer son analyse en détail
- ❖ Le développement logiciel dépend de l'utilisation de code d'autres équipes
- ❖ En Java, la Javadoc est utilisée avec les annotations de documentation :
 - ❖ `@param @return @throws` : méthodes
 - ❖ `@author @version` : classes
 - ❖ `@see @since` : si nécessaire
- ❖ Toutes les classes, toutes les méthodes publiques doivent être documentées